

Project Background

1 Overview

OSTicket is open source web based support ticket system build in PHP and uses MySQL as database. It also supports email and API. The current project involves building up an add-on (Plugin) for OSTicket that can send SMS notifications along with email for the notifications settings on OSTicket.

The first phase includes development of add-on for sending SMS notifications using OSTicket module.

1.1 Features

Following are the high level features of the OSTicket application:

1. It supports custom fields.
2. Rich Text HTML
3. Ticket Filters
4. Help Topics
5. Agent Collision Avoidance
6. Assign and Transfer
7. Auto Responder
8. Internal Notes
9. Service Level Agreements
10. Customer Portal
11. Dashboard Portals.

1.2 Technology Stack

Requirement was to make the application using following technology:

1. PHP ver. 5.3 and above
2. MySQL Database

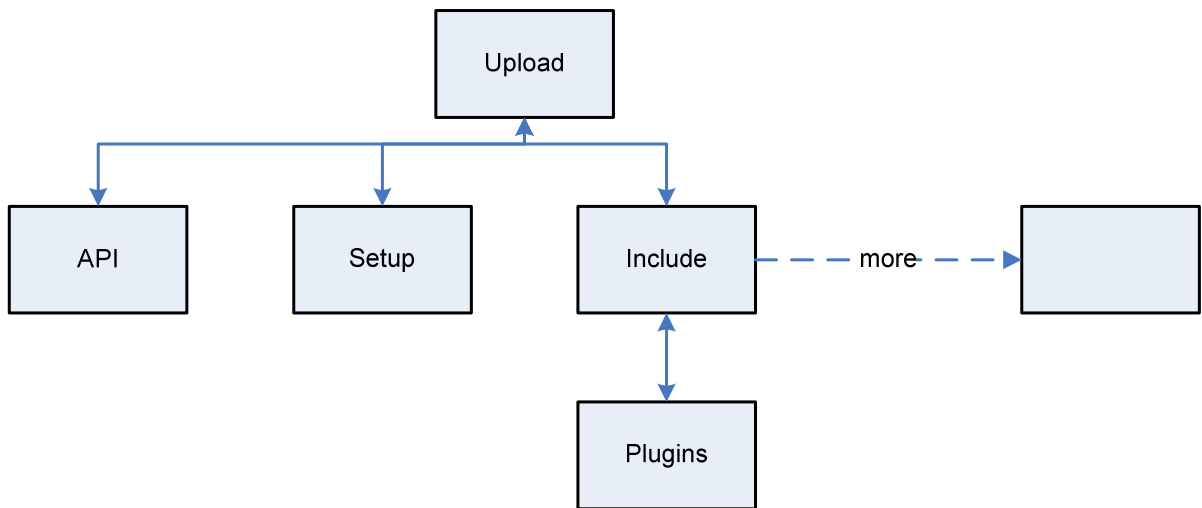
2 Application Design

Following is how the application is designed.

OSTicket contains 2 folders.

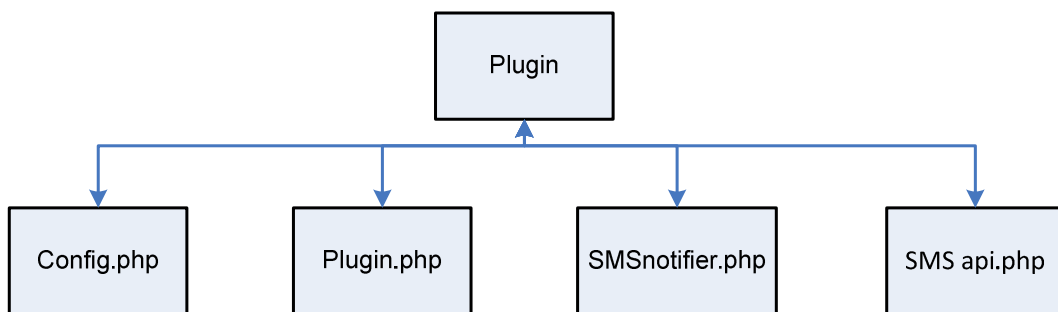
1. Upload
2. Scripts

- Upload is the main directory for the installation and needs to be placed in the root directory for the OSTicket installation.
- Scripts contain helper scripts for the OSTicket.
- Upload has folders that perform different tasks for the OSTicket project few of which are described below.
- Setup Folder: The folder contains files that are used for the setup of the support system.
- API Folder: Contains files and folders that provide API support for the OSTicket System.
- Include Folder: Contains helper libraries and plugins.



3 Development of the Plugin

Plugin will require all the files for the plugin located in a single folder preferably with the name of the Plugin in our case SMSNotifier.



A plugin requires a set of files that are necessary for the definition of the same named below.

1. Config.php : this file contains the configuration information for the plugin for eg. In our case the API endpoint for sending SMS etc.
2. Plugin.php: this file returns an array containing information on the plugin for eg. Version number for the plugin.
3. SMSNotifier.php: This class would contain the code for the plugin including functions to send SMS etc.
4. Sms api.php: This class would be a wrapper to call the API functions from sms api.

Config.php

This would contain a class with the name SMSNotifierConfig.php that would extend class PluginConfig as standard.

Within this class there would be a function getOptions that would return the options required for the plugin.

There would be another function pre_save that would validate the settings or options before saving them.

Plugin.php

This file returns the array with information for the plugin like name and version number etc.

SMSNotifier.php

This class would contain generic function like constructor and sendSMS that would send SMS using the gateway selected from the options using settings for the gateway.

Sms api.php

This class would be used if the sms gateway is selected from options and would connect to the gateway using credentials saved in the settings and call the wrapper function to send SMS.

These SMS's have to be binded with what we know in OSTicket architecture as Signals.

Signals are the events that we can use and bind the custom functions with the events.

Signals are defined in class Signal defined in include/class.signal.php and has 2 functions defined connect and send.

Connect is used to subscribe to a signal and send is used to publish a signal.

We may use signal that are generated while sending email notifications so that we can bind to these and also send the SMS notifications along with email based on the configuration saved.